

L^AT_EX Schmerzfrei

Martin Girschick¹

Philipp Matthias Hahn²

November 1997 für das Seminar Softwareentwicklungswerkzeuge

¹<mailto:girschik@igd.fhg.de>

²<mailto:pmhahn@igd.fhg.de>

Inhaltsverzeichnis

1	Einführung	4
1.1	Was ist \LaTeX ?	4
1.2	Was bietet \LaTeX ?	4
1.3	Woher bekommt man \LaTeX ?	5
1.4	Wie bedient man \LaTeX ?	5
2	Strukturiertes Layout mit \LaTeX	7
2.1	Die \LaTeX Dokument-Klassen	7
2.1.1	Anwendung des Klassenkonzepts	7
2.1.2	Die Standard-Klassen von \LaTeX	7
2.1.3	Optionale Parameter	7
2.2	Gliederungselemente	8
2.2.1	Titelseite	8
2.2.2	Gliederung des Textes	8
2.3	Inhaltsverzeichnisse	8
2.4	Bibliographien	8
2.5	Querverweise	9
3	Gestaltungsmöglichkeiten	10
3.1	Textformatierungen	10
3.1.1	Schriftschnitt und Schriftgröße	10
3.1.2	New Font Selection Scheme	11
3.1.3	Die Änderungen von $\text{\LaTeX} 2_{\epsilon}$	12
3.2	Absatzformatierungen	12
3.2.1	Vertikalausrichtung	12
3.2.2	Strukturierte Absätze	12
3.2.3	Horizontaler Zwischenraum	13
3.2.4	Gruppierungen	13
3.2.5	Absatzboxen	13
4	\LaTeX-Objekte	14
4.1	Tabellen	14
4.1.1	Die tabbing-Umgebung	14
4.1.2	Die tabular-Umgebung	14
4.2	Listen	15
4.2.1	itemize	15
4.2.2	enumerate	16
4.2.3	description	16
4.3	Formeln	16

4.4	Grafiken	17
4.5	Gleitende Umgebungen	17
4.6	Fußnoten	18
A	Symboltabellen	19
A.1	mathematische Symbole	19
A.1.1	Mathematische Akzente	19
A.1.2	Kleine griechische Buchstaben	19
A.1.3	Große griechische Buchstaben	19
A.1.4	„Große“ Operatoren	20
A.1.5	Binäre Operatoren	20
A.1.6	Relationen	20
A.1.7	Pfeile	21
A.1.8	Klammern	21
A.1.9	Verschiedene sonstige Symbole	21
A.2	Sonderzeichen im Text-Modus	21
B	Befehle	22
B.1	Textformatierungen	22
B.2	Absatzformatierungen	22
B.3	Befehle zum Aufbau eines L ^A T _E X-Dokuments	23
B.4	Tabellenbefehle	23
B.5	Aufzählungen	23
B.6	Formeln	23
	Literaturverzeichnis	23
	Nachwort	24

Kapitel 1

Einführung

1.1 Was ist \LaTeX ?

\LaTeX ¹ ist ein Makropaket für \TeX ² das von Leslie Lamport entwickelt wurde. Durch dieses Makropaket ist es möglich, recht schnell durchstrukturierte Dokumente zu erstellen, die, ohne das man viel tun muß, ein vernünftiges Aussehen besitzen. Vor allem die Erstellung von immer gleich aussehenden Dokumenten, eine im wissenschaftlichen Bereich oft gehörte Forderung, wird so erleichtert.

\LaTeX existiert in zwei Versionen: Das ursprüngliche \LaTeX wurde im Laufe der Zeit bis Version 2.09 weiterentwickelt: Durch die weltweite Fortentwicklung traten jedoch immer mehr Inkompatibilitäten zwischen den Erweiterungen auf. Deshalb wurden große Teile neuimplementiert, was leider auch dazu führt, das die beiden Versionen nicht miteinander kompatibel sind. Die neue Version wird als $\LaTeX 2_{\epsilon}$ bezeichnet und ist nun wesentlich flexibler und gestattet es, einzelne Pakete einzubinden, die dann die entsprechende Funktionalität implementieren.

Für das vorliegende Dokument wurde als Aussehen der Book-Stil gewählt. Dadurch haben die Kopf- und Fußzeilen ein vordefiniertes Aussehen. Das Inhaltsverzeichnis wurde durch den Befehl `\tableofcontents` eingefügt.

1.2 Was bietet \LaTeX ?

\LaTeX ermöglicht es, durch sogenannte *Dokument-Klassen* Schriftstücken ein fest vorgegebenes Aussehen zu verleihen. In diesen Klassen werden z.B. die Seiteneinstellungen, Ränder, Schriftarten und Formate definiert. In dem eigentlichen Dokument wird dann nur noch durch sogenannte *Markups* auf z.B. Überschrift umgeschaltet. Das eigentliche Aussehen wird erst durch Definitionen der Klassen festgelegt. Da das Makropaket weiß, was eine Überschrift ist, ist es ein leichtes, automatische Inhaltsverzeichnisse zu generieren. Auch der Formelsatz, der größtenteils aus \TeX -Elementen besteht, ist von großem Nutzen. Komplexe Konstrukte wie Inhaltsverzeichnisse, Bibliographien, Indizes, Tabellen und viele mehr werden unterstützt und führen durch die automatische Generierung sehr schnell zu ansehnlichen Ergebnissen.

¹gesprochen lah-tech

² \TeX ist ein Textsatzsystem, das von Donald Knuth entwickelt wurde.

1.3 Woher bekommt man L^AT_EX ?

Da T_EX für fast alle Rechnerplattformen und Betriebssysteme verfügbar ist, stellt die Verwendung von L^AT_EX kein Problem dar. Ein Netzwerk namens CTAN³, stellt weltweit verschiedene Server bereit, auf denen alle öffentlich vertriebenden Zusätze für T_EX und L^AT_EX zu finden sind. Diese Server gleichen sich automatisch aufeinander ab, so daß alle Server den gleichen Stand haben. Die „Deutsche Anwendervereinigung T_EX e.V.“, kurz DANTE, betreibt einen dieser Rechner, der über das Internet unter `ftp://ftp.dante.de` erreichbar ist.

1.4 Wie bedient man L^AT_EX ?

L^AT_EX arbeitet wie ein Compiler: Man schreibt zunächst einen Quelltext, der die Formatierungsanweisungen und natürlich den eigentlichen Text enthält. Dazu kann man jeden Editor verwenden, der eine reine ASCII-Datei schreiben kann. Es gibt spezielle Editoren⁴, die einen bei der Eingabe von T_EX-Dateien unterstützen und etwa für häufig benötigte Sprachkonstrukte Assistenten anbieten oder einfach auch nur die Schlüsselwörter farbig hervorheben. Es hat sich eingebürgert, als Endung für den Quelltext „.tex“ zu wählen.

Unter Unix wird L^AT_EX z.B. mit `latex beispiel.tex` aufgerufen. Wenn man keinen Fehler gemacht hat, wird eine Datei `beispiel.dvi` erzeugt, die dann mittels `dvips beispiel.dvi` in ein PostScript-Dokument übersetzt werden kann oder aber mittels `xdvi beispiel.dvi` auf einer grafikfähigen Unix-Konsole angezeigt werden kann⁵.

L^AT_EX liest die Eingabedatei sequentiell ein, verarbeitet die Befehle und macht einige Statusausgaben. Trifft er bei der Verarbeitung auf Fehler, so zeigt er eine mehr oder minder sinnvolle Fehlermeldung an und wartet auf eine Eingabe. Mit `<Return>` kann man meistens den Fehler übergehen und L^AT_EX zur Weiterarbeit überreden. Die Fehler sollte man dann im Quelltext verbessern und L^AT_EX erneut starten, bis man eine fehlerfreie Version hat. In den meisten Implementationen ist es möglich, mit `x <Return>` die Verarbeitung abubrechen oder mit `q <Return>`⁶ keine weiteren Fehlermeldungen anzuzeigen⁷.

Benutzt man in seinem Dokument Inhaltsverzeichnisse, Indizes oder Querverweise, kann es nötig sein, L^AT_EX bis zu drei mal aufzurufen, um korrekte Seitenzahlen zu erhalten.

Eine L^AT_EX-Datei sollte folgendes Aussehen haben:

```
\documentclass [Optionen] {Klassenname}

\usepackage [german] {babel}      % falls benoetigt
\usepackage [latin1] {inputenc} % Umlauten direkt im Quellcode

\begin{document}
...
\end{document}
```

Wie in T_EX werden Spaces, Tabulatoren und Returns als Leerzeichen interpretiert. Absätze erhält man, indem man eine Leerzeile einfügt.

³Comprehensive T_EX Archive Network

⁴z.B. Erweiterungen für Emacs oder für den Macintosh den Editor Alpha Lite

⁵Für nahezu alle Plattformen stehen auch Frontends zur Verfügung, die die Bedienung etwas erleichtern.

⁶run quietly

⁷Diese werden jedoch alle in die Datei `beispiel.log` geschrieben.

Die im Text eingestreuten Befehle beginnen meist mit einem Backslash gefolgt von dem eigentlichen Befehl. Der Befehl wird mit einem Space oder einem anderen Nichtbuchstaben abgeschlossen. Um nach dem Befehl ein Leerzeichen einzufügen, muß der Befehl mit `{ }` enden. Diese geschweiften Klammern enthalten unter Umständen auch die nötigen Parameter für den Befehl. Je nach Befehl können optionale Parameter in eckigen Klammern angegeben werden.

Aufbau eines Befehls `\Befehlsname[optionale Parameter]{Parameter}`

Umgebungen Konstrukte, die von `\begin{Bezeichner}` und `\end{Bezeichner}` eingeschlossen sind bezeichnet man als Umgebungen. Der oben dargestellt Eingabetext befindet sich zum Beispiel in einer `verbatim`-Umgebung. Diese ist zur Darstellung von solchem Eingabetext gedacht.

Deutschsprachige Dokumente Um \LaTeX -Dokumente in anderen Sprachen außer Englisch zu verfassen empfiehlt sich das Babel-Paket. Durch `\usepackage[Sprachename]{babel}` wird die Sprachenunterstützung aktiviert. Es stehen dann eine Reihe von Befehlen zur Verfügung, mit denen man zwischen den verschiedenen Sprachen und dazugehörigen Trennmustern umschalten kann.

Verwendet man `german` als Sprachename, so wird die richtige Trenndatei verwendet und es stehen folgende Kommandos zur Verfügung:

Umlaute Das aus \TeX bekannte vorangestellte `\` wurde zu `"` vereinfacht (`"U` \rightarrow `Ü` etc.)

Scharfes S Kann nun durch `"s` (\rightarrow `ß`) erzeugt werden

Explizite Trennstellen erzeugt man durch `"-`, um Trennstellen zu deaktivieren verwendet man `"|`

Anführungszeichen „ und “ werden durch `"` gefolgt von `'` (Accent grave) bzw. `'` (Accent aigu) erzeugt.

Kapitel 2

Strukturiertes Layout mit L^AT_EX

2.1 Die L^AT_EX Dokument-Klassen

2.1.1 Anwendung des Klassenkonzepts

Jedes L^AT_EX-Dokument beginnt mit der Anweisung:

```
\documentclass[Optionen]{Klasse}
```

Man wählt damit ein bestimmtes Aussehen für das Dokument aus. Dieses wird in den `.cls`-Dateien definiert¹.

2.1.2 Die Standard-Klassen von L^AT_EX

In einer Standard-Installation von L^AT_EX sind normalerweise die Klassen `article`, `report`, `book`, `proc`, `slides`² und `letter`³ enthalten.

2.1.3 Optionale Parameter

Die oben genannten Klassen lassen sich noch durch diverse Parameter steuern. Diese müssen durch Komma getrennt in dem Feld `options` angegeben werden. Die Schriftgröße kann von von der Standard-Einstellung 10pt durch Angabe von 11pt oder 12pt verändert werden.

Desweiteren existieren die Optionen `a4paper`, `landscape`, `twocolumn` und `twoside`. Letzteres sorgt für einen alternierenden Seitenspiegel für doppelseitigen Druck (Standard bei `book`).

Für deutschsprachige Dokumente kann man durch `\usepackage[german]{babel}` ein entsprechendes Erweiterungspaket laden.⁴ Damit wird die deutsche Trenntabelle aktiviert und die englischen Überschriften durch die deutschen Bezeichnungen wie „Kapitel“, „Inhaltsverzeichnis“, „Index“ und „Literaturverzeichnis“ ersetzt. Zusätzlich werden auch noch die Datumsausgabe auf Deutsch und das Papierformat auf DIN A4 umgestellt.

Weitere Pakete für die unterschiedlichsten Anwendungsbereiche sind bei den bekannten Quellen erhältlich und in vielen Distributionen bereits enthalten.

¹Eigene Klassen können auch erstellt werden, wie man an den verfügbaren Erweiterungen für L^AT_EX sieht.

²Für Folien ist FoilTeX sehr nützlich, jedoch nicht in allen Distributionen verfügbar (Klasse `foils`)

³Für DIN-Briefe existieren entsprechende Anpassungen

⁴Babel stellt auch Anpassungen für andere Sprachen zur Verfügung (zur Erstellung mehrsprachiger Texte).

2.2 Gliederungselemente

2.2.1 Titelseite

Die Titelseite dieses Dokuments wurde durch den folgenden Eingabetext erzeugt:

```
\title{\LaTeX\ Schmerzfrei}
\author{Martin Girschick\thanks{mailto:girschik@igd.fhg.de}
\and Philipp Matthias Hahn\thanks{mailto:pmhahn@igd.fhg.de}}
\date{November~1997 f"ur das Seminar Softwareentwicklungswerkzeuge}
\maketitle
```

Wie man sieht, wird durch `\thanks{text}` eine Fußnote⁵ erzeugt. Durch `\maketitle` wird dann der Titel gesetzt. Wenn man `\date{...}` wegläßt, wird das Datum gemäß der aktuellen Sprache eingesetzt.

2.2.2 Gliederung des Textes

Zur Gliederung des Textes stehen die Befehle `part`, `chapter`⁶, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph` zur Verfügung. Je nach Dokumentenklasse werden die Gliederungen automatisch in das Inhaltsverzeichnis eingefügt. In "Book" oder „Report" sind dies z.B. `chapter`, `section` und `subsection`. Als Parameter übergibt man die entsprechenden Kapitelüberschriften.

Beispiel `\subsection{Gliederung des Textes}`

Durch das Einfügen des Befehls `\appendix` wird der Anhang eingeleitet. Dies führt unter Anderem dazu, daß die Kapitelnummerierung auf Buchstaben umgestellt wird.

2.3 Inhaltsverzeichnisse

Wie bereits in 1.1 erwähnt wird das Inhaltsverzeichnis durch den Befehl `\tableofcontents` erzeugt. Da sich nach dem Einfügen des Inhaltsverzeichnisses noch Seitenzahlen ändern können, ist es unter Umständen nötig, daß Dokument mehrmals zu „T_EXen“.

2.4 Bibliographien

Literaturverzeichnisse⁷ erhält man in L^AT_EX durch eine `thebibliography`-Umgebung. Dort sollte für jeden Verweis ein `bibitem`-Eintrag erstellt werden.

Aufbau `\bibitem{Kurzbezeichnung} Text`

⁵Wie Fußnoten im laufenden Text angebracht werden, wird später in 4.6 beschrieben.

⁶steht in `article` nicht zur Verfügung

⁷mit dem Paket `BIBTEX` können auch komplexe Datenbanken mit Bibliographien verwaltet werden.

Beispiel

```

\begin{thebibliography}{99}
  \bibitem{kopka} Helmut Kopka \\\
                  {\LaTeX\ --- Eine Einf"uhrung} \\\
                  Addison--Wesley (Deutschland) GmbH, 4.~Auflage~1992
  ...
\end{thebibliography}

```

Die Angabe „99“ hinter der Umgebungseröffnung dient als Muster für die Breite der Nummerierung, die jedem Eintrag vorangestellt wird.

Will man dann eines dieser Werke zitieren, so genügt ein `\cite[Text]{Kurzbezeichnung}`. In dem Feld *Text* kann noch optional Text eingefügt werden, der dann nach dem Verweis erscheint, um auf bestimmte Seiten zu verweisen. Weitere Informationen dazu findet man in [2, Seite 56-57] (Erzeugt durch den Eingabetext `\cite[Seite 56-57]{kopka}`).

2.5 Querverweise

Querverweise innerhalb eines Dokumentes werden mit einer `label`-Struktur erzeugt. An der Verweisstelle fügt man mit `\label{Bezeichnung}` ein Label ein. Dabei ist darauf zu achten, daß das Label direkt hinter der Überschrift bzw. der `caption`-Zeile der Grafik bzw. Tabelle steht. Um auf dieses Label zu referenzieren verwendet man dann `\ref{Bezeichnung}`. Um lediglich auf die Seitennummer zu verweisen, kann man stattdessen `\pageref{Bezeichnung}` benutzen. Bei vielen Referenzen empfiehlt sich das *varioref*-Paket, daß eine Mischung der beiden Standardreferenzen bietet: Bei Referenzen wird automatisch ein Text entsprechend der Entfernung benutzt, also etwa „Bild 3 auf der nächsten Seite“ oder „Tabelle 2.7 auf Seite 13“.

Kapitel 3

Gestaltungsmöglichkeiten

L^AT_EX bietet als Textsatzsystem natürlich auch Möglichkeiten, das Aussehen des Textes anzupassen. Im Gegensatz zu anderen Textverarbeitungen geht es dabei nicht um das *wie*, sondern um das *was*. Man gibt nicht explizit an, daß die Überschrift als 12 Punkte, fett, Times Roman gesetzt werden soll, sondern man gibt lediglich an, daß es sich bei diesem Text um eine Überschrift handelt. In den Dokumentklassen sind dann das Aussehen in entsprechenden Makrodefinitionen festgelegt. Dieses System bietet den Vorteil, das sich der Autor ganz auf den Text konzentrieren kann und sich nicht als Setzer zu versuchen braucht.

Durch diese Vorgehensweise erreicht man eine Entkoppelung von Text und Satz: Ändert man z.B. die Standardschriftgröße im Vorspann auf 11 Punkte, so werden alle Zeichen im Text entsprechend geändert. Hätte man genau angegeben, wie ein Zeichen zu formatieren ist, so müßte man den gesamten Text durchgehen und jedes Zeichen einzeln anpassen. Dokumente lassen sich so sehr schnell an andere Wünsche anpassen. So haben Verlage oft spezielle Vorstellung vom Aussehen und bieten ihre eigenen Klassen an.

Ein weiterer Vorteil ergibt sich dadurch, daß man diese Deklarationen auch dazu verwenden kann, automatisch Inhaltsverzeichnisse und andere Strukturen zu erstellen, wie bereits in Kapitel 2.3 erwähnt wurde. Manchmal möchte man jedoch von der Automatisierung abweichen und Texte bzw. Absätze besonders *hervorheben*.

3.1 Textformatierungen

3.1.1 Schriftschnitt und Schriftgröße

Zur Hervorhebung von Text steht zum Beispiel `\emph{text}` zur Verfügung. Je nach Stil wird dieser z.B. in kursiv oder fett gesetzt. Diese Möglichkeit reicht normalerweise auch schon aus, aber sowohl die Schriftgröße als auch der Schriftschnitt lassen sich ohne größere Probleme verändern. Die dazu nötigen Befehle werden im Folgenden vorgestellt. Wie bei Befehlen üblich werden sie mit Backslash eingeleitet, gefolgt von dem Befehlsname und dem zu setzenden Text in geschweiften Klammern.

Folgende Möglichkeiten stehen zur Verfügung:

- **fett mit `textbf`**
- *kursiv mit `textit`*
- underline mit `underline`
- KAPITÄLCHEN MIT `TEXTSC`

- Schreibmaschinenschrift mit `texttt`
- Serifenlose Schrift mit `textsf`

Schriftgrößen lassen sich auf dieselbe Weise verändern¹:

`tiny` `scriptsize` `footnotesize` `small` `normalsize` `large` `Large` `LARGE` `huge` `Huge`

3.1.2 New Font Selection Scheme

Bei gleichzeitiger Anwendung von Schriftselektierung und Größenauswahl kann eine der beiden Deklarationen verlorengehen. Ab den Versionen 3 von `TEX` bzw. bei `LATEX 2ε` gibt es ein neues Zeichensatz-Auswahlverfahren², wo man getrennt die Schriftfamilie (`family`), die Stärke (`series`), die Form (`shape`) und Größe (`size`) angeben kann.

Schriftfamilien Die Familie bezeichnet die Herkunft der Schrift, wie z.B. `cmr`³ für eine Schrift mit Serifen, `cmss` für eine serifenlose Schrift und `cmtt` für eine Schreibmaschinenschrift. Um eine Familie auszuwählen, benutzt man in `LATEX` das Makro `\family{fam}`.

Schriftstärken Die Stärke gliedert sich in zwei Teilattribute, die einerseits die Liniendicke und andererseits die Zeichenbreite angeben. Die Dicke läßt sich von UltraLight (`ul`) über Light (`l`), Medium (`m`), Bold (`b`) bis hinzu UltraBold (`ub`) variieren. Für die Weite gibt es ähnliche Bezeichnungen, wie UltraCondensed (`uc`), Condensed (`c`), Medium (`m`), Expanded (`e`) bis zu UltraExpanded (`ux`). Gesetzt wird dieses Attribut durch `\series{st_br}`, wobei sich `st_br` aus der Stärke und Weite zusammensetzt. Eine sehr dünne aber gedehnte Schrift wäre also `ulx`⁴.

Schriftschnitt Über den Schnitt kann man zwischen normaler (`n`), italic (`it`), geneigter (`sl`) und Kapitälchen (`sc`) sowie einer ungeneigten Italicschrift (`u`) wählen. Gesetzt werden diese Attribute mit `\shape{Schnitt}`.

Schriftgröße und Zeilenabstand Mit `\size{Größe}{Zeilenabstand}` legt man schließlich die Größe der Schrift in Druckerpunkten sowie den Zeilenabstand zwischen zwei Zeilen fest.

Verwendung Sobald die Werte gesetzt sind, kann der Zeichensatz mit `\selectfont` aktiviert werden. In den neueren Versionen von `LATEX` wurden die oben beschriebenen Fontauswahl- und Schriftgrößenbefehle an die gerade beschriebene Umgebung angepasst, so daß z.B. ein `\textrm` gleichbedeutend ist mit `\family{cmr}\selectfont`.

¹In `LATEX 2ε` gibt es auch die Möglichkeit, stattdessen `Größendefinition{Text}` zu schreiben.

²NFSS, von *New Font Selection Scheme*

³Computer Modern Roman

⁴`ul` von UltraLight und `x` von Expanded

3.1.3 Die Änderungen von $\text{\LaTeX} 2_{\epsilon}$

In $\text{\LaTeX} 2_{\epsilon}$ haben sich die Namen der Makros leicht geändert: Alle Tags fangen jetzt mit `\text` an, um die Funktion der Markups deutlicher zu machen. Statt `\rm` schreibt man jetzt also besser `\textrm` für die Auswahl der Roman-Schrift. Der zu formatierende Text wird in geschweiften Klammern dahinter angegeben. Soll nur die aktuelle Umgebung verändert werden, heißen die Tags wieder anders. Für genaue Informationen siehe [7, Seite 67].

So schön die Möglichkeiten zur expliziten Schriftauswahl auch sind, sollte man es nicht übertreiben: Viele verschiedene Schriften verwirren den Leser meistens nur und lenken vom eigentlichen Inhalt ab. Durch die direkte Auswahl geht auch der Vorteil verloren, daß sich die Änderung eines Stils auf das gesamte Dokument auswirkt. Die direkte Zeichensatzauswahl sollte man eigentlich nur dann benutzen, wenn man eigene Stylesheets erzeugen will oder wirklich zwingend eine bestimmte Formatierung benötigt.

3.2 Absatzformatierungen

Wie bereits in 1.4 erwähnt wird Eingabetext von \LaTeX nach folgenden Regeln behandelt: Mehrere Leerzeichen werden zu einem zusammengefaßt. Ein einzelner Zeilenumbruch ist gleichbedeutend mit einem Leerzeichen. Zwei Absätze werden durch eine Leerzeile getrennt.

3.2.1 Vertikalausrichtung

Innerhalb eines Absatzes kann man jedoch auch eine neue Zeile beginnen, indem man als Zeilentrenner `\`, `\newline` bzw. `*[Abstand]`⁵ benutzt. Die dritte Form fügt zusätzlichen Platz zwischen den Zeilen ein. Dieses kann man auch durch `\smallskip`, `\medskip` und `\bigskip` erreichen, die jeweils 1/4, 1/2 oder einen ganze Zeilenhöhe Platz schaffen. Beliebiger Platz kann mit `\vspace*{Abstand}` eingerichtet werden.

Um einen Seitenumbruch zu erzwingen, fügt man an der Stelle ein `\newpage` oder `\clearpage` ein. Bei einem zweiseitigen Satz kann man mit `\cleardoublepage` automatisch eine weitere Leerseite einfügen, damit der nächste Text auf der rechten Seite beginnt. Um einen Seitenumbruch zu verhindern existiert die `\samepage`-Umgebung. Darin enthaltener Text wird nur zwischen Absätzen umgebrochen.

Bei Befehlen, in denen ein `*` auftritt, wird der Abstand auf jeden Fall eingefügt, auch wenn dieser am Anfang einer Seite steht. Es existiert eine weitere Form dieser Befehle ohne den Stern, die diesen Leerraum automatisch unterdrückt.

3.2.2 Strukturierte Absätze

\LaTeX rückt in einen Absatz die erste Zeile leicht ein, außer es ist der erste Absatz nach einer Überschrift. Um diese Regel zu durchbrechen, kann man direkt vor einem Absatz `\indent` bzw. `\noindent` schreiben.

\LaTeX setzt einen Absatz normalerweise in Blocksatz. Möchte man statt dessen den Text links- oder rechtsbündig schreiben, ist dies durch Einschließen in eine `flushleft`- bzw. `flushright`-Umgebung möglich. Einen zentrierten Absatz schreibt man dagegen zwischen `\begin{center}` und `\end{center}`.

Es gibt auch noch spezielle Umgebungen, um z.B. Zitate (`quote`), Gedichte (`verse`) oder vorformatierte Texte (`verbatim`) zu schreiben. Letztere wird z.B. dafür benutzt, um \LaTeX -

⁵Als Maßeinheiten existieren Zentimeter (cm), Millimeter (mm), Inches (in), Punkte (pt), Picas (1pc=12pt), em (Breite des Gedankenstrichs) sowie ex (Höhe des Buchstabens x).

Eingabetext darzustellen⁶. Will man die Leerzeichen sichtbar machen, so heißt die Umgebung `verbatim*`. Um ganze Dateien einzufügen, bietet sich das `verbatim`-Paket an: Mit `\verbatiminput{dateiname}` kann man z.B. den Quelltext seines Programmes in die Dokumentation einfügen und ist so das Problem der Inkonsistenz von Dokumentation und Programm los. Die `verbatim`-Umgebung hat den Nachteil, daß der Inhalt nicht umgebrochen wird und deshalb in eine Zeile passen muß. Das `path`-Paket bietet hierzu das `\path|...|`-Konstrukt, das ähnlich, wie `\verb|...|` arbeitet, jedoch den Inhalt notfalls trennt und umbricht.

3.2.3 Horizontaler Zwischenraum

Neben der Auswahl von verschiedenen Zeichensätzen kann man aber auch die standardmäßige Zwischenraumautomatik manuell überschreiben. `\enspace`, `\quad` und `\qquad` erzeugen einen halben, einen ganzen bzw. zwei ganze Leerräume⁷. Mit `\hspace*{Leerraum}` kann man einen beliebigen horizontalen Zwischenraum einfügen. Mit `\hrulefill{Leerraum}` erhält man alternativ einen Leerraum mit einer durchgezogenen Linie oder mit `\dotfill{Leerraum}` einer gepunkteten Linie. Um ein geschütztes Leerzeichen⁸ zu erzeugen, ersetzt man es durch eine Tilde (`~`).

3.2.4 Gruppierungen

Text, der nicht durch einen Zeilenumbruch getrennt werden soll, kann durch eine `\mbox{Text}` zu einer Einheit zusammengefaßt werden. Um Text oder eine Box zu umranden, kann dieser in eine `\fbox{...}` eingefügt werden. Durch das mehrfache Einschachteln in eine `fbox` bekommt man auch einen doppeltem Rahmen.

3.2.5 Absatzboxen

Um Textabschnitte nebeneinander zu setzen, rahmt man die Texte jeweils mit einer Box ein. Dazu gibt es mehrere Möglichkeiten:

- `\parbox[position]{breite}{text}`
- `\begin{minipage}[position]{breite} Text`
`\end{minipage}`

Beide werden wie ein einzelnes Zeichen behandelt. Mit `position` kann man angeben, ob die Box relativ zum verbleibenden Text der gleichen Zeile oben (t) oder unten (b) bündig abschließen soll. Wird die Positionsangabe weggelassen, wird die Box zentriert. Durch geschickte Wahl der Breite und dem Einfügen von horizontalem Platz kann man so Text nebeneinander setzen. So ist z.B. das Beispiel auf Seite 14 mittels folgendem Konstrukt gesetzt:

```
\begin{minipage}{7cm} ... \end{minipage}
\hfill
\begin{minipage}{9cm} ... \end{minipage}
```

⁶Für kürzere Texte ist `\verb|Text|` besser geeignet. Anstatt des Pipe-Zeichens kann man auch jedes andere Zeichen verwenden.

⁷Ein Leerraum entspricht der Breite des Buchstabens „M“ in der aktuellen Schrift.

⁸geschützte Leerzeichen setzt man zwischen Wörter, wenn diese nicht am Zeilenende auseinandergerissen werden sollen.

Kapitel 4

L^AT_EX-Objekte

4.1 Tabellen

4.1.1 Die `tabbing`-Umgebung

Tabellen sind eines der schwierigsten Gebiete des Textsatzes. Deshalb bietet hier L^AT_EX mehrere Konstrukte an, mit denen man Tabellen erzeugen kann. Einfache Tabellen kann man mit den Tabulatorfunktionen¹ innerhalb einer `tabbing`-Umgebung erzeugen. Mit `\=` wird ein Tabulator an der entsprechenden horizontalen Position gesetzt, der dann später mit `\>` angefahren werden kann. Die einzelnen Zeilen werden durch ein `\\` abgeschlossen. Tabulatoren sind normalerweise alle links ausgerichtet. Um Text an einem Tabulator rechts auszurichten, wird der Text zwischen `\>` und `\'` gesetzt². Soll ein Text ganz an den rechten Seitenrand gesetzt werden, setzt man vor den Text ein `\'`. Danach darf kein weiterer Tabulator folgen.

Beispiel

Spaltentitel 1	Titel 2	rechte Spalte
Anfang	Ende	
rechts	links	rechts

```
\begin{tabbing}
Spaltentitel 1 \= Titel 2 \= rechte Spalte \\
Anfang        \>           \> Ende           \\
\>rechts      \\' links   \> rechts         \\
\end{tabbing}
```

4.1.2 Die `tabular`-Umgebung

Einfache Tabellen mit gleichbleibender Spaltenzahl und -breite lassen sich mit der `tabular`-Umgebung erzeugen. Die Vorteile zur `tabbing`-Umgebung liegen vor allem in der Möglichkeit, horizontale und vertikale Striche einzufügen.

Aufbau

```
\begin{tabular}{Formatierung}
\hline          % Horizontale Linie, falls gewünscht
Text & Text ... \\
...
\hline          % Horizontale Linie, falls gewünscht
\end{tabular}
```

¹Eine ähnliche Funktionsweise ist von der Schreibmaschine bekannt.

²`\'` trennt eigentlich zwei Textteile. Der Text vor `\'` wird rechts- und der Text danach linksbündig gesetzt

Bei Formatierung muß für jede Spalte die Ausrichtung durch l, c, r (respektive Linksbündig, Zentriert, Rechtsbündig) bestimmt werden. Um die Spalten durch vertikale Linien zu trennen, muß man an den entsprechenden Stellen ein oder zwei | einfügen, um eine einfache bzw. doppelte Trennungslinie zu erzeugen.

Beispiel

Zentriert	Links	Rechts	Zentriert
1	2	3	4
Langer Text	b	$\sum_{i=1}^{\infty} \frac{1}{i} > x \forall x$	c

```

\begin{tabular}{|c|lr|c|}
\hline
Zentriert & Links & Rechts & Zentriert \\
\hline\hline
1 & 2 & 3 & 4 \\
Langer Text & b &  $\sum_{i=1}^{\infty} \frac{1}{i} > x \forall x$  & c \\
\hline
\end{tabular}

```

An dem Beispiel erkennt man, das die Spaltenbreite automatisch an die breiteste Zelle der Spalte angepaßt wird. Anstatt der Ausrichtung kann man jedoch auch mit `p{Breite}` die Breite explizit angeben. Dadurch kann man auch Fließtext in eine Zelle einfügen, der dann korrekt umgebrochen wird. Will man benachbarte Spalten zu einer Spalte zusammenfassen oder die Ausrichtung einer einzelnen Zelle ändern, ist `\multicolumn{Spaltenanzahl}{Format}{Text}` sehr nützlich.³

4.2 Listen

In $\text{\LaTeX}2_{\epsilon}$ gibt es drei Grundtypen von Listen, die die wichtigsten Anwendungsbereiche abdecken.

Aufbau

```

\begin{Bezeichnung}
\item Text
...
\item Text
\end{Bezeichnung}

```

4.2.1 itemize

Je nach Schachtelungstiefe wird bei dieser unnummerierten Liste ein ausgefüllter Kreis, ein Gedankenstrich, ein Stern oder ein kleiner Punkt als Gliederungssymbol ausgegeben. Eigene Symbole können mit `\item[Symbol]` verwendet werden.

³Weitere Pakete wie `dcolumn`, `array`, `longtable` für komplexere Tabellen sind verfügbar.

4.2.2 enumerate

Je nach Tiefe der Aufzählung werden hier arabische Zahlen, kleine Buchstaben, römische Ziffern oder Großbuchstaben als Gliederungsmerkmal benutzt.

4.2.3 description

Diese Liste ist gut für Definitionen geeignet: Nach `item` kann man in eckigen Klammern einen Begriff angeben, der dem eigentlichen Text fett vorangestellt wird.

Beispiel	<code>\begin{description}</code>
	<code>\item[Listen] in \LaTeX</code>
Listen in L ^A T _E X	<code>\begin{enumerate}</code>
1. itemize	<code>\item itemize</code>
2. enumerate	<code>\item enumerate</code>
3. description	<code>\item description</code>
	<code>\end{enumerate}</code>
Tabellen in L ^A T _E X	<code>\item[Tabellen] in \LaTeX</code>
	<code>\begin{itemize}</code>
• tabular	<code>\item tabular</code>
• table	<code>\item table</code>
	<code>\end{itemize}</code>
	<code>\end{description}</code>

4.3 Formeln

Formeln im laufenden Text werden durch `$` eingeschlossen. Verwendet man stattdessen eine `displaymath`-Umgebung, wird die Formel vom laufenden Text abgesetzt. In der `equation`-Umgebung wird zusätzlich an den rechten Rand eine fortlaufende Nummer gesetzt. Der Eingabetext für die $a^2 + b^2 = c^2$ ist `$a^{2} + b^{2} = c^{2}$`.

Ein weiteres Beispiel zeigt die Verwendung von Summenzeichen⁴:

$$\sum_{x=1}^n x = \frac{n(n+1)}{2} \quad (4.1)$$

Der entsprechende Eingabetext:

```
\begin{equation}
\sum_{x=1}^n x = \frac{n(n+1)}{2}
\end{equation}
```

Beispiel mit einer eqnarray-Umgebung

$$(1) \quad zero \neq succ(x) \quad \forall x \in N \quad (4.2)$$

$$(2) \quad succ(x) = succ(y) \Rightarrow x = y \quad \forall x, y \in N \quad (4.3)$$

(3) für jede Teilmenge $P \subseteq N$ gilt :

$$P = N \Leftrightarrow a) zero \in P$$

$$b) succ(P) \subseteq P,$$

$$\text{d.h. } \forall n \in N. n \in P \rightarrow succ(n) \in P \quad (4.4)$$

⁴weitere Symbole können aus der Kurzreferenz entnommen werden.


```

\begin{eqnarray}
(1) & \& \text{zero} \neq \text{succ}(x) & & \& \text{forall } x \in N \ \backslash\backslash \\
(2) & \& \text{succ}(x) = \text{succ}(y) \Rightarrow x=y & & \& \text{forall } x,y \in N \ \backslash\backslash \\
(3) & \& \text{"ur jede Teilmenge } P \subseteq N \text{ gilt: } & \backslash\backslash \\
& \& P = N \Leftrightarrow \text{: a) } \text{zero} \in P & \backslash\backslash \\
& \& \hspace*{2.5cm} \text{b) } \text{succ}(P) \subseteq P, & \backslash\backslash \\
& \& \text{"d.h.} \text{: } \text{forall } n \in N. n \in P \Rightarrow \text{succ}(n) \in P & \backslash\backslash \\
\end{eqnarray}

```

Wie man sieht, kann man mit `hspace` Abstände einfügen, mit `mbox` wird normaler Text eingefügt. Zeilen werden mit Doppel-Backslash abgeschlossen, ein vorangestelltes `\nonumber` verhindert die Ausgabe der Formelnummer. `\:` dient dazu, kleine Abstände einzufügen.

Wenn man sehr viele Formeln braucht, ist $\mathcal{A}_{\mathcal{M}\mathcal{S}}\text{-}\text{\LaTeX}$ sehr zu empfehlen. Dieses Paket wurde von der „American Mathematical Society“ entworfen, um einige Restriktionen von \LaTeX auszubessern. Zusätzlich wurden noch etliche Sonderzeichen hinzugefügt, die vorher nicht enthalten waren.

4.4 Grafiken

Grafiken sind in fast allen Textverarbeitungsprogrammen ein Problem. Auch \LaTeX macht hier nicht unbedingt eine Ausnahme: Schon die Tatsache, daß eine geräteunabhängige Datei erzeugt wird, macht das Einbinden von Grafiken nicht leichter. In den meisten \LaTeX -Distributionen sind Pakete zum Einbinden von Postscript-Dateien⁵ (z.B. `\includegraphics{picture}`) vorhanden. Weitere Varianten sind meist Plattformspezifisch, Informationen hierzu findet man dann in der Anleitung der jeweiligen \LaTeX -Implementation.

Für einfache Liniengrafiken besteht die Möglichkeiten, direkt mit \LaTeX zu arbeiten: Neben Linien, Pfeilen, Rechtecken, Kreisen und abgerundeten Kästen kann man in diesen Grafiken alle anderen Fähigkeiten wie Formelsatz und Textsatz nutzen. Diese Möglichkeiten sind aber eingeschränkt, da \LaTeX diese Grafiken durch das geschickte Zusammenfügen von verschiedenen Zeichen aus verschiedenen Zeichensätzen darstellt. So sind für Linien nur bestimmte Neigungen zulässig und die Radien der Kreise dürfen nur bestimmte Werte annehmen. Da das Erstellen solcher Grafiken sehr mühsam ist, existieren Programme, wie z.B. `xfig` für Unix, die spezielle Modi bieten, um um Grafiken im \LaTeX -Format zu erstellen. Solche Grafiken können entweder direkt im Dokument stehen oder aber mit `\input{Dateiname}` aus einer externen Datei gelesen werden.

4.5 Gleitende Umgebungen

Werden Tabellen, Grafiken oder Formeln in Dokumenten benutzt, passen diese oft nicht mehr auf die Seite, an der sie eingefügt wurden. \LaTeX kann solche Objekte beweglich halten, um das Seitenlayout zu verbessern (keine großen Löcher). Für Tabellen existiert die `table`-Umgebung, für Grafiken die `figure`-Umgebung.

Aufbau

```

\begin{Umgebungsname}[Position]
  \caption{Beschreibung}\label{Name} % label, falls noetig
  ...

```

⁵die Ausgabe ist dann nur auf einem Postscript-Drucker oder einem -Interpreter (Ghostscript etc.) möglich

```
\end{Umgebungsname}
```

Je nachdem, ob die Caption-Zeile vor oder nach dem Text erscheint, wird sie über oder unter den Text gesetzt.

Für Position gilt:

h (here) Das Objekt wird möglichst an genau der eingefügten Stelle plaziert.

t (top) Das Objekt wird bevorzugt an den oberen Rand der aktuellen, ggf. auch nächsten Seite gesetzt.

b (bottom) Die Ausrichtung erfolgt am unteren Rand der Seite.

p (page) Die Umgebung wird auf einer eigenen Seite dargestellt.

4.6 Fußnoten

Fußnoten dienen dazu, Anmerkungen im Text anzubringen, die Begriffe genauer erklären oder zusätzliche Informationen bieten. Oft werden sie auch im Zusammenhang mit Zitaten benutzt, wenn der Autor eigene Anmerkungen zum zitierten Text machen möchte. Fußnoten werden von L^AT_EX jeweils fortlaufend nummeriert. Je nach Dokumentenklasse wird die Zählung kapitelweise (`book`, `report`) oder fortlaufend (`article`) durchgeführt⁶.

Aufbau und Verwendung an der zu referenzierenden Stelle⁷

```
\footnote[Symbol]{Text}
```

⁶Weitere Pakete wie `footnpag` ermöglichen beispielsweise seitenweise Numerierung.

⁷In einigen Versionen von L^AT_EX muß man statt `footnote` `cite` verwenden.

Anhang A

Symboltabellen

A.1 mathematische Symbole

In den folgenden Tabellen (entnommen aus [5]) sind alle Symbole angeführt, die im mathematischen Modus verwendet werden können. Bei vielen Installationen stehen mit den Paketen `latexsym`, `amssymb`, `mathrsfs` oder `wasysym` weitere Zeichen zur Verfügung.

A.1.1 Mathematische Akzente

<code>\hat a</code>	\hat{a}	<code>\check a</code>	\check{a}	<code>\tilde a</code>	\tilde{a}	<code>\acute a</code>	\acute{a}
<code>\grave a</code>	\grave{a}	<code>\dot a</code>	\dot{a}	<code>\ddot a</code>	\ddot{a}	<code>\breve a</code>	\breve{a}
		<code>\bar a</code>	\bar{a}	<code>\vec a</code>	\vec{a}		

A.1.2 Kleine griechische Buchstaben

α	<code>\alpha</code>	ι	<code>\iota</code>	ϱ	<code>\varrho</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	σ	<code>\sigma</code>
γ	<code>\gamma</code>	λ	<code>\lambda</code>	ς	<code>\varsigma</code>
δ	<code>\delta</code>	μ	<code>\mu</code>	τ	<code>\tau</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	υ	<code>\upsilon</code>
ε	<code>\varepsilon</code>	ξ	<code>\xi</code>	ϕ	<code>\phi</code>
ζ	<code>\zeta</code>	o	<code>o</code>	φ	<code>\varphi</code>
η	<code>\eta</code>	π	<code>\pi</code>	χ	<code>\chi</code>
θ	<code>\theta</code>	ϖ	<code>\varpi</code>	ψ	<code>\psi</code>
ϑ	<code>\vartheta</code>	ρ	<code>\rho</code>	ω	<code>\omega</code>

A.1.3 Große griechische Buchstaben

Γ	<code>\Gamma</code>	Ξ	<code>\Xi</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	Π	<code>\Pi</code>	Ψ	<code>\Psi</code>
Θ	<code>\Theta</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>
Λ	<code>\Lambda</code>	Υ	<code>\Upsilon</code>		

A.1.4 „Große“ Operatoren

Σ	\sum	<code>\sum</code>	\cap	\bigcap	<code>\bigcap</code>	\odot	\bigodot	<code>\bigodot</code>
Π	\prod	<code>\prod</code>	\cup	\bigcup	<code>\bigcup</code>	\otimes	\bigotimes	<code>\bigotimes</code>
\amalg	\coprod	<code>\coprod</code>	\sqcup	\bigsqcup	<code>\bigsqcup</code>	\oplus	\bigoplus	<code>\bigoplus</code>
\int	\int	<code>\int</code>	\vee	\bigvee	<code>\bigvee</code>	\uplus	\biguplus	<code>\biguplus</code>
\oint	\oint	<code>\oint</code>	\wedge	\bigwedge	<code>\bigwedge</code>			

A.1.5 Binäre Operatoren

$+$	$+$	<code>+</code>	$-$	$-$	<code>-</code>		
\pm	\pm	<code>\pm</code>	\cap	\cap	<code>\cap</code>	\vee	<code>\vee</code>
\mp	\mp	<code>\mp</code>	\cup	\cup	<code>\cup</code>	\wedge	<code>\wedge</code>
\setminus	\setminus	<code>\setminus</code>	\uplus	\uplus	<code>\uplus</code>	\oplus	<code>\oplus</code>
\cdot	\cdot	<code>\cdot</code>	\sqcap	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>
\times	\times	<code>\times</code>	\sqcup	\sqcup	<code>\sqcup</code>	\otimes	<code>\otimes</code>
$*$	$*$	<code>\ast</code>	\triangleleft	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
\star	\star	<code>\star</code>	\triangleright	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\diamond	\diamond	<code>\diamond</code>	\wr	\wr	<code>\wr</code>	\dagger	<code>\dagger</code>
\circ	\circ	<code>\circ</code>	\bigcirc	\bigcirc	<code>\bigcirc</code>	\ddagger	<code>\ddagger</code>
\bullet	\bullet	<code>\bullet</code>	\triangleup	\triangleup	<code>\triangleup</code>	\amalg	<code>\amalg</code>
\div	\div	<code>\div</code>	\triangledown	\triangledown	<code>\triangledown</code>		

A.1.6 Relationen

Die meisten der Symbole können durch ein vorangestelltes *not*-Befehl auch in negierter Form dargestellt werden.

$<$	$<$	<code><</code>	$>$	$>$	<code>></code>	$=$	$=$	<code>=</code>
\leq	\leq	<code>\leq</code>	\geq	\geq	<code>\geq</code>	\equiv	\equiv	<code>\equiv</code>
\prec	\prec	<code>\prec</code>	\succ	\succ	<code>\succ</code>	\sim	\sim	<code>\sim</code>
\preceq	\preceq	<code>\preceq</code>	\succeq	\succeq	<code>\succeq</code>	\simeq	\simeq	<code>\simeq</code>
\ll	\ll	<code>\ll</code>	\gg	\gg	<code>\gg</code>	\asymp	\asymp	<code>\asymp</code>
\subset	\subset	<code>\subset</code>	\supset	\supset	<code>\supset</code>	\approx	\approx	<code>\approx</code>
\subseteq	\subseteq	<code>\subseteq</code>	\supseteq	\supseteq	<code>\supseteq</code>	\cong	\cong	<code>\cong</code>
\sqsubseteq	\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	\bowtie	<code>\bowtie</code>
\in	\in	<code>\in</code>	\ni	\ni	<code>\ni</code>	\propto	\propto	<code>\propto</code>
\vdash	\vdash	<code>\vdash</code>	\dashv	\dashv	<code>\dashv</code>	\models	\models	<code>\models</code>
\smile	\smile	<code>\smile</code>	\mid	\mid	<code>\mid</code>	\doteq	\doteq	<code>\doteq</code>
\frown	\frown	<code>\frown</code>	\parallel	\parallel	<code>\parallel</code>	\perp	\perp	<code>\perp</code>

A.1.7 Pfeile

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Uparrow	<code>\Uparrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>				

A.1.8 Klammern

$($	<code>(</code>	$)$	<code>)</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>
$[$	<code>[</code>	$]$	<code>]</code>	$\{$	<code>\{</code>	$\}$	<code>\}</code>
\lbrack	<code>\lbrack</code>	\rbrack	<code>\rbrack</code>	$\{$	<code>\lbrace</code>	$\}$	<code>\rbrace</code>

A.1.9 Verschiedene sonstige Symbole

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\exists	<code>\exists</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\neg	<code>\neg</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\natural	<code>\natural</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\sharp	<code>\sharp</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\clubsuit	<code>\clubsuit</code>	\clubsuit	<code>\clubsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	\diamondsuit	<code>\diamondsuit</code>	\diamondsuit	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>	\spadesuit	<code>\spadesuit</code>

A.2 Sonderzeichen im Text-Modus

\dagger	<code>\dagger</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>
$\#$	<code>\#</code>	\P	<code>\P</code>	$_$	<code>_</code>
$\$$	<code>\\$</code>	$\&$	<code>\&</code>	$\%$	<code>\%</code>
$-$	<code>-</code>	$-$	<code>--</code>	$-$	<code>---</code>

Anhang B

Befehle

B.1 Textformatierungen

Schriften und ihre Eigenschaften

ART	DICKE	NEIGUNG
roman (<code>\textrm</code>)	fett (<code>\textbf</code>)	schräg (<code>\textsl</code>)
Schreibmaschine (<code>\texttt</code>)	medium (<code>\textmd</code>)	kursiv (<code>\textit</code>)
serifenlos (<code>\textsf</code>)		aufrecht (<code>\textup</code>)
		Kapitalchen (<code>\textsc</code>)

Unterstützte Größen

<code>\tiny</code>	winzig
<code>\scriptsize</code>	Indexgröße
<code>\footnotesize</code>	Fußnotengröße
<code>\small</code>	klein
<code>\normalsize</code>	normale Textgröße
<code>\large</code>	groß
<code>\Large</code>	sehr groß
<code>\LARGE</code>	sehr sehr groß
<code>\huge</code>	riesig
<code>\Huge</code>	monströs

B.2 Absatzformatierungen

```
\verb/.../  
\clearpage \cleardoublepage  
\indent \noindent  
\begin{flushright} ... \end{flushright}  
\begin{flushleft} ... \end{flushleft}  
\begin{center} ... \end{center} \centerline{...}  
\begin{quote} ... \end{quote}  
\begin{verse} ... \end{verse}  
\smallskip \medskip \bigskip \vspace*  
\enspace \quad \qquad \hspace*
```

B.3 Befehle zum Aufbau eines L^AT_EX-Dokuments

```

\documentclass [Optionen] {Klasse}
\usepackage [german] {babel}
\begin{document} ... \end{document}

\begin{thebibliography} ... \end{thebibliography}
\bibitem {Kurzbezeichnung} Text
\cite [Text] {Kurzbezeichnung}
\label {Bezeichnung}
\ref {Bezeichnung}
\footnote [Symbol] {Text}

\part {Titel}      \chapter {Titel}
\section {Titel}   \subsection {Titel}
\paragraph {Titel} \subparagraph {Titel}
\appendix

\begin{minipage} {Breite} ... \end{minipage}

```

B.4 Tabellenbefehle

```

\begin{tabbing} ... \end{tabbing}
\= % settab
\> % jumptotab
\\ % newline

\begin{tabular} {Format} ... \end{tabular}
\hline % horizontal line

```

B.5 Aufzählungen

```

\begin{itemize} ... \end{itemize}
\begin{enumerate} ... \end{enumerate}
\begin{description} ... \end{description}
\item [Symbol]

```

B.6 Formeln

```

$...$
\begin{displaymath} ... \end{displaymath}
\begin{equation} ... \end{equation}
\begin{eqnarray} ... \end{eqnarray}

```

Literaturverzeichnis

- [1] Helmut Kopka
L^AT_EX — Eine Einführung
Addison–Wesley (Deutschland) 1988
- [2] Helmut Kopka
L^AT_EX — Eine Einführung
Addison–Wesley (Deutschland) GmbH, 4. Auflage 1992
- [3] Dr. Reinhard Wonneberger
Kompaktführer L^AT_EX
Addison–Wesley (Deutschland) GmbH, 2. Auflage 1990
- [4] Fritz Cremer
Das kleine T_EXBuch
1993
- [5] Jörg Knappen, Hubert Partl, Elisabeth Schlegl, Irene Hyna
L^AT_EX 2_ε-Kurzbeschreibung
Version 1.2, 12. November 1995
- [6] Leslie Lamport
Das L^AT_EX–Handbuch
Addison–Wesley (Deutschland) GmbH, 1995
- [7] Christine Detig
Der L^AT_EX Wegweiser
Thomson Publishing, 1. Auflage 1997
- [8] Deutsche Anwendervereinigung für T_EX e.V.
<http://www.dante.de>

Nachwort Diese kurze L^AT_EX -Einführung entstand im Rahmen des Seminars „Software–Entwicklungswerkzeuge – Literate Programming“ (WS97/98). An dieser Stelle möchten wir Herrn Dr.-Ing. K. Guntermann, welcher das Seminar leitete, für seine Unterstützung bei der Erstellung des Dokuments danken.

Martin Girschick und Philipp Matthias Hahn, Dezember 1997